

***DISASTER RECOVERY DENGAN PROXMOX HA CLUSTER
DAN FREENAS UNTUK MENAMBAH AVAILABILITY SERVER***



**Disusun sebagai salah satu syarat menyelesaikan Program Studi Strata I pada Jurusan
Informatika Fakultas Komunikasi dan Informatika**

Oleh:

ARDITYA DEVRIAN ALBAR

L 200 120 124

**PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2016**

HALAMAN PERSETUJUAN

***DISASTER RECOVERY DENGAN PROXMOX HA CLUSTER
DAN FREENAS UNTUK MENAMBAH AVAILABILITY SERVER***

PUBLIKASI ILMIAH

L 200 120 124

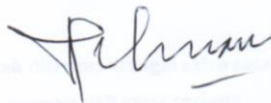
oleh: **Ardaya Devrian Albar**
NPM.120120124
Jurusan Informatika
Fakultas Ilmu Komputer
Universitas Indonesia
Jakarta, 15 Mei 2024

ARDITYA DEVRIAN ALBAR

L 200 120 124

Telah diperiksa dan disetujui untuk diuji oleh:

Dosen Pembimbing



Helman Muhammad, S.T., M.T.

NIK.1564

HALAMAN PENGESAHAN

**DISASTER RECOVERY DENGAN PROXMOX HA CLUSTER
DAN FREENAS UNTUK MENAMBAH AVAILABILITY SERVER**

OLEH

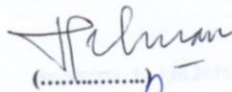
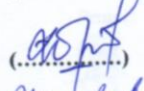
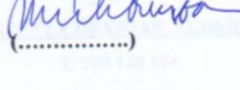
ARDITYA DEVRIAN ALBAR

L 200 120 124

Telah dipertahankan di depan Dewan Penguji
Fakultas Komunikasi Dan Informatika
Universitas Muhammadiyah Surakarta
Pada hari Senin .. 15 .. Agustus 2016
dan dinyatakan telah memenuhi syarat

Dewan Penguji:

1. Helman Muhammad, S.T., M.T.
(Ketua Dewan Penguji)
2. Aris Rakhmadi, S.T., M.Eng.
(Anggota I Dewan Penguji)
3. Muhammad Kusban, S.T., M.T.
(Anggota II Dewan Penguji)


(.....)

(.....)

(.....)

Publikasi ilmiah ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar sarjana

Tanggal 13 Agustus .. 2016

Mengetahui,

Dekan
Fakultas Komunikasi dan Informatika

Husni Thamrin, S.T., M.T., Ph.D.
NIK : 706

Ketua Program Studi
Informatika

Dr. Heru Supriyono, M.Sc.
NIK:970

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam publikasi ilmiah ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 15 Juli 2016

Penulis



ARDITYA DEVRIAN ALBAR

L 200 120 124



**UNIVERSITAS MUHAMMADIYAH SURAKARTA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
PROGRAM STUDI INFORMATIKA**

Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id>. Email: informatika@ums.ac.id

SURAT KETERANGAN LULUS PLAGIASI

012/A.3-IL3/INF-FKI/VII/2016

Assalamu'alaikum Wr. Wb

Biro Skripsi Program Studi Informatika menerangkan bahwa :

Nama : ARDITYA DEVRIAN ALBAR
NIM : L200120124
Judul : DISASTER RECOVERY DENGAN PROXMOX HA CLUSTER
DAN FREENAS UNTUK MENAMBAH AVAILABILITY SERVER
Program Studi : Informatika
Status : Lulus

Adalah benar-benar sudah lulus pengecekan plagiasi dari Naskah Publikasi Skripsi, dengan menggunakan aplikasi Turnitin.

Demikian surat keterangan ini dibuat agar dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Wr. Wb

Surakarta, 28 Juli 2016

Biro Skripsi Informatika

Endang Wahyu Pamungkas, S.Kom., M.Kom.

SCAN Turnitin

Turnitin - Google Chrome

https://turnitin.com/newreport.asp?r=48.525528836673&svr=05&lang=en_us&oid=692077502&pbd=2&ft=1

preferences

previous paper next paper

turnitin

Originality Report

Document Viewer

Processed on: 27-Jul-2016 13:17 WIB

ID: 692077502

Word Count: 2110

Submitted: 1

DISASTER RECOVERY

DENGAN PROXMOX

HA CLUSTER D...

By Ardit Devrian

Similarity Index

14%

Similarity by Source

Internet Sources:	10%
Publications:	0%
Student Papers:	9%

exclude quoted exclude bibliography exclude small matches

mode: show highest matches together

DISASTER RECOVERY DENGAN PROXMOX HA CLUSTER DAN FREENAS UNTUK MENAMBAH AVAILABILITY SERVER Abstrak

Server merupakan sebuah sistem komputer yang bertugas sebagai penyedia layanan untuk client. Dengan fungsi server yang

memberikan layanan kepada client, maka server dituntut untuk

mempunyai tingkat availability yang tinggi. Akan tetapi pasti akan terdapat masalah yang dapat menyebabkan server tidak dapat memenuhi tugasnya sebagai penyedia layanan, entah itu sebuah bencana alam atau bencana yang disebabkan human error. Untuk itu perlu adanya disaster recovery untuk mengatasi hal tersebut. Penelitian ini bertujuan untuk merancang disaster recovery dengan membuat sistem cluster computer yang dibangun dari dua buah server virtual dengan sistem operasi Proxmox VE 3.4. Sistem tersebut dibuat untuk dapat meminimalisir downtime server dan menambah tingkat availability server. Selain menggunakan cluster computer untuk mendukung perancangan disaster recovery, juga dibutuhkan sebuah metode share storage yang digunakan sebagai data center yang dibangun dari satu buah server virtual dengan sistem operasi FreeNAS. Berdasarkan penelitian didapatkan hasil yang hampir mendekati zero downtime. Dalam pengujian yang dilakukan beberapa kali didapat rata-rata delay waktu 0.39 menit dengan tingkat availability total mencapai 99.32 %.

sehingga sistem cluster virtual ini dapat menjadi solusi untuk meningkatkan sistem dengan tingkat availability yang tinggi. Kata Kunci: Disaster Recovery, Availability, Cluster

Computer, Share Storage Abstract The server is a computer system that served as a service provider for the client. With

1

4% match (student papers from 06-Jul-2015)

Class publikasi

Assignment publikasi september 2015

Paper ID: [554227503](#)

2

2% match (student papers from 22-Jul-2016)

Class Publikasi Wisuda Juni

Assignment Publikasi Wisuda September 2016

Paper ID: [691068273](#)

3

1% match (Internet from 11-Nov-2013)

<http://news.palcomtech.com>

4

1% match (Internet from 19-Jul-2016)

<https://www.scribd.com/doc/310237801/Nasi-Publikasi-Sistem-Pakar-Untuk-Pertolongan-Pertama-Gangguan-Kesehatan-Pada-Anak>

5

1% match (Internet from 07-Oct-2015)

<http://eprints.ums.ac.id>

6

1% match (student papers from 09-Jul-2015)

Class publikasi

DISASTER RECOVERY DENGAN PROXMOX HA CLUSTER DAN FREENAS UNTUK MENAMBAH AVAILABILITY SERVER

Abstrak

Server merupakan sebuah sistem komputer yang bertugas sebagai penyedia layanan untuk *client*. Dengan fungsi *server* yang memberikan layanan kepada *client*, maka *server* dituntut untuk mempunyai tingkat *availability* yang tinggi. Akan tetapi pasti akan terdapat masalah yang dapat menyebabkan *server* tidak dapat memenuhi tugasnya sebagai penyedia layanan, entah itu sebuah bencana alam atau bencana yang disebabkan *human error*. Untuk itu perlu adanya *disaster recovery* untuk mengatasi hal tersebut. Penelitian ini bertujuan untuk merancang *disaster recovery* dengan membuat sistem *cluster computer* yang dibangun dari dua buah *server virtual* dengan sistem operasi Proxmox VE 3.4. Sistem tersebut dibuat untuk dapat meminimalisir *downtime server* dan menambah tingkat *availability server*. Selain menggunakan *cluster computer* untuk mendukung perancangan *disaster recovery*, juga dibutuhkan sebuah metode *share storage* yang digunakan sebagai *data center* yang dibangun dari satu buah *server virtual* dengan sistem operasi FreeNAS. Berdasarkan penelitian didapatkan hasil yang hampir mendekati *zero downtime*. Dalam pengujian yang dilakukan beberapa kali didapat rata-rata *delay waktu* 0.39 menit dengan tingkat *availability* total mencapai 99.32 %, sehingga sistem *cluster virtual* ini dapat menjadi solusi untuk meningkatkan sistem dengan tingkat *availability* yang tinggi.

Kata Kunci: *Disaster Recovery, Availability, Cluster Computer, Share Storage*

Abstract

The server is a computer system that served as a service provider for the client. With server functions which provide services to the client, the server is required to have a high level of availability. But surely there will be a problem that can cause the server able to fulfill his duties as a service provider, whether it's a natural disaster or a disaster caused by human error. For that we need a disaster recovery to overcome it. This research aims to design a disaster recovery by creating a cluster computer system built from two virtual servers with the operating system Proxmox VE 3.4. The system is built to minimize server downtime and increase the level of availability server. In addition to using a computer cluster to support the design of disaster recovery, also needed a method of shared storage that is used as a data center built on a single virtual server with FreeNAS operating system. Based on the research results obtained were almost close to zero downtime. In tests conducted several times gained an average delay time of 0.39 minutes with a total availability rate reached 99.32%, so the virtual cluster system this can be a solution to improve the system with a high level of availability.

Keywords: *Disaster Recovery, Availability, Cluster Computer, Share Storage*

1. PENDAHULUAN

Server merupakan sebuah sistem komputer yang bertugas sebagai penyedia layanan untuk *client*. Server diharapkan dapat selalu berada dalam keadaan aktif dan perangkat lunak dan proses yang ada dalam server berjalan secara normal (Thamrin, 2011). Dengan fungsi *server* yang

memberikan layanan kepada *client*, maka *server* dituntut untuk mempunyai tingkat *availability* yang tinggi. Pada zaman teknologi maju seperti sekarang ini *server* tidak hanya terbatas pada *server* fisik saja tetapi sudah ada teknologi virtualisasi *server*. Dengan teknologi virtualisasi kita dimudahkan dalam hal *maintenance server*, pengadaan fisik *server*, dan biaya yang dikeluarkan lebih murah.

Akan tetapi pasti akan terdapat masalah yang dapat menyebabkan *server* tidak dapat memenuhi tugasnya sebagai penyedia layanan, entah itu sebuah bencana alam atau bencana yang disebabkan *human error*. Untuk itu perlu adanya *disaster recovery* untuk mengatasi hal tersebut. Salah satu teknologi dalam virtualisasi yang dapat digunakan dalam perancangan *disaster recovery* adalah *cluster computer*. *Cluster computer* merupakan sebuah teknologi dimana terdapat dua atau lebih komputer yang dihubungkan (*cluster/clustering*) untuk dapat bersama-sama mengerjakan tugas yang diberikan. Dengan metode tersebut dapat meminimalisir *downtime server* dan menambah tingkat *High Availability server*. Berdasarkan penelitian yang dilakukan oleh Irfani (2015) diperoleh nilai *availability* paling besar mencapai 99,50% dengan tingkat kestabilan *cluster* dari sisi *CPU utilization* dan *Throughput*, sehingga sistem *cluster virtual* ini dapat menjadi solusi untuk meningkatkan sistem dengan tingkat *availability* yang tinggi.

Selain menggunakan *cluster computer* untuk mendukung perancangan *disaster recovery*, juga dibutuhkan sebuah metode *share storage* yang digunakan sebagai *data center*. Sehingga data dapat disimpan terpusat dalam *server* yang berbeda.

2. METODE

Penelitian ini bertujuan untuk merancang *disaster recovery* dengan membuat sistem *cluster computer* untuk dapat meminimalisir *downtime server* dan menambah tingkat *availability server*. Dari penelitian ini akan didapatkan hasil berupa nilai *availability server*. Dalam penelitian ini dilakukan pengujian dengan mematikan salah satu *server* yang menjalankan *service* dan mengamati *service* akan berpindah ke *server* backup serta menghitung waktu *downtime server* untuk mendapatkan nilai *availability server*.

2.1 Waktu dan Tempat

Waktu yang digunakan untuk menyelesaikan penelitian ini sekitar 6 bulan yaitu bulan November 2015 sampai bulan Mei 2016 yang dilakukan di UMS.

2.2 Alat dan Bahan

Peralatan utama dalam penelitian ini dibagi menjadi dua kategori yaitu perangkat keras (*hardware* dan perangkat lunak (*software*). *Hardware* yang digunakan adalah Laptop Asus A451LN dengan sistem operasi Windows 8.1 Pro x64 dan spesifikasi Processor Intel® Core™ i5-4200U CPU @ 1.60 GHz 2.30 GHz, Harddisk 1TB dan RAM 8GB. Software yang digunakan

untuk penelitian adalah *Virtualbox*, *Proxmox VE*, *FreeNAS*, *Web Browser (Mozilla Firefox / Google Chrome)*.

2.3 Perancangan dan Implementasi Server

a. Instalasi Virtual Server

Dalam menyiapkan *server* diperlukan aplikasi *Virtualbox* untuk membuat *virtual server* dengan spesifikasi *virtual server* 1 dan 2 dengan *OS Proxmox* seperti pada Tabel 1.

Tabel 1. Spesifikasi Virtual Server 1 dan 2.

Spesifikasi	Virtual Server	
	Server 1	Server 2
RAM	1024 MB	1024 MB
Harddisk	8 GB	8 GB
Network Adapter	Host Only Adapter	Host Only Adapter
IP Address	192.168.1.14	192.168.1.13

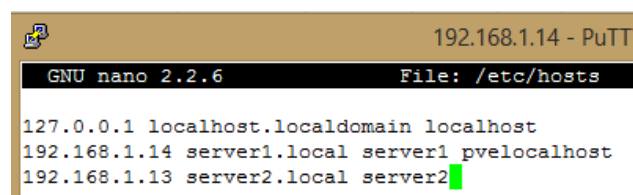
Dan untuk spesifikasi *virtual server* 3 dengan *OS FreeNAS* seperti pada Tabel 2.

Tabel 2. Spesifikasi virtual server 3

Hardware	Kapasitas	Keterangan
RAM	1024 MB	
Harddisk	<ul style="list-style-type: none"> 2 GB 20 GB 	<ul style="list-style-type: none"> System NFS
Network Adapter	Host Only Adapter	IP 192.168.1.12

b. Konfigurasi membuat *cluster*

Masuk pada **/etc/hosts** untuk menambahkan *host proxmox* pada kedua *server*. Pada *host server* 1, dengan mengetikkan *ip address* dan *hostname server* 2, kemudian pada *server* 2 mengetikkan *ip address* dan *hostname server* 1.



```

192.168.1.14 - PuTTY
GNU nano 2.2.6 File: /etc/hosts
127.0.0.1 localhost.localdomain localhost
192.168.1.14 server1.local server1 pvelocalhost
192.168.1.13 server2.local server2

```

Gambar 1. Penambahan *IP address* dan *hostname server* 2 pada *server* 1

```

192.168.1.13 - PuTTY
GNU nano 2.2.6      File: /etc/hosts
127.0.0.1 localhost.localdomain localhost
192.168.1.13 server2.local server2 pvelocalhost
192.168.1.14 server1.local server1

```

Gambar 2. Penambahan *IP address* dan *hostname* server 1 pada server 2

Kemudian membuat *cluster* dengan mengetikkan perintah pada server 1 **pvecm create koneksi** lalu berpindah ke server 2 untuk menambahkan anggota *cluster* pada server 1 dengan mengetikkan **pvecm add 192.168.1.14**

c. Konfigurasi *fencing*

Konfigurasi *fencing* bertujuan agar *node* tidak mengakses data yang sama saat terjadi *failover cluster* sehingga mengurangi terjadinya data rusak. Tahap pertama adalah dengan meng-copy file **cluster.conf** dengan perintah **cp /etc/pve/cluster.conf /etc/pve/cluster.conf.new** kemudian mengedit file **cluster.conf.new** seperti pada Gambar 3.

```

1  <?xml version="1.0"?>
2  <cluster name="koneksi" config_version="3">
3
4      <cman two_node="1" expected_votes="1" keyfile="/var/lib/pve-cluster/corosync.authkey">
5      </cman>
6
7  <fencedevices>
8      <fencedevice agent="fence_ilo" ipaddr="192.168.1.14" login="root" passwd="g00gl345" name="fenceA"/>
9      <fencedevice agent="fence_ilo" ipaddr="192.168.1.13" login="root" passwd="g00gl345" name="fenceB"/>
10 </fencedevices>
11
12 <clusternodes>
13     <clusternode name="server1" nodeid="1" votes="1">
14         <fence>
15             <method name="1">
16                 <device name="fenceA" action="reboot"/>
17             </method>
18         </fence>
19     </clusternode>
20     <clusternode name="server2" nodeid="2" votes="1">
21         <fence>
22             <method name="1">
23                 <device name="fenceB" action="reboot"/>
24             </method>
25         </fence>
26     </clusternode>
27 </clusternodes>
28
29 </cluster>

```

Gambar 3. Konfigurasi *fencing* pada **cluster.conf.new**

Kemudian mengedit file **/etc/default/redhat-cluster-pve** pada kedua server Proxmox. *Enable*-kan pada konfigurasi **FENCE_JOIN="yes"** dengan menghilangkan tanda pagar. Lalu mengaktifkan *fence_tool* pada kedua server Proxmox dengan perintah **fence_tool join**.

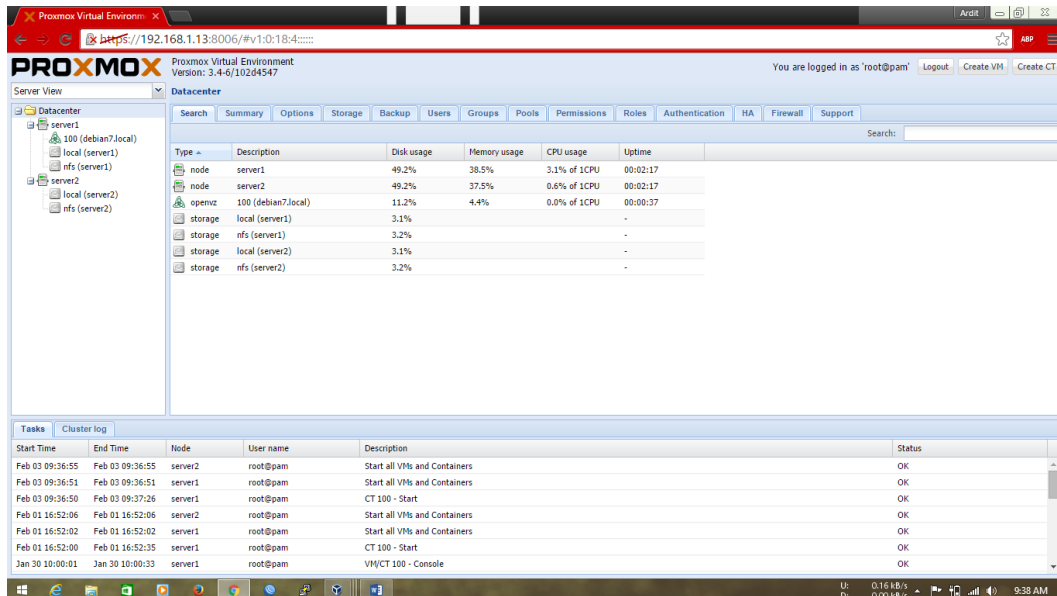
3. HASIL DAN PEMBAHASAN

3.1 Hasil Penelitian

Setelah melakukan pembuatan sistem, langkah selanjutnya adalah pengujian sistem. Pengujian sistem dilakukan dengan men-*shutdown* atau me-*reboot* salah satu *node* atau *server* dan kemudian melihat apakah *service* yang berjalan pada *server* yang dimatikan tersebut akan berpindah secara otomatis atau *failover* ke *node* yang aktif (*High Availability*).

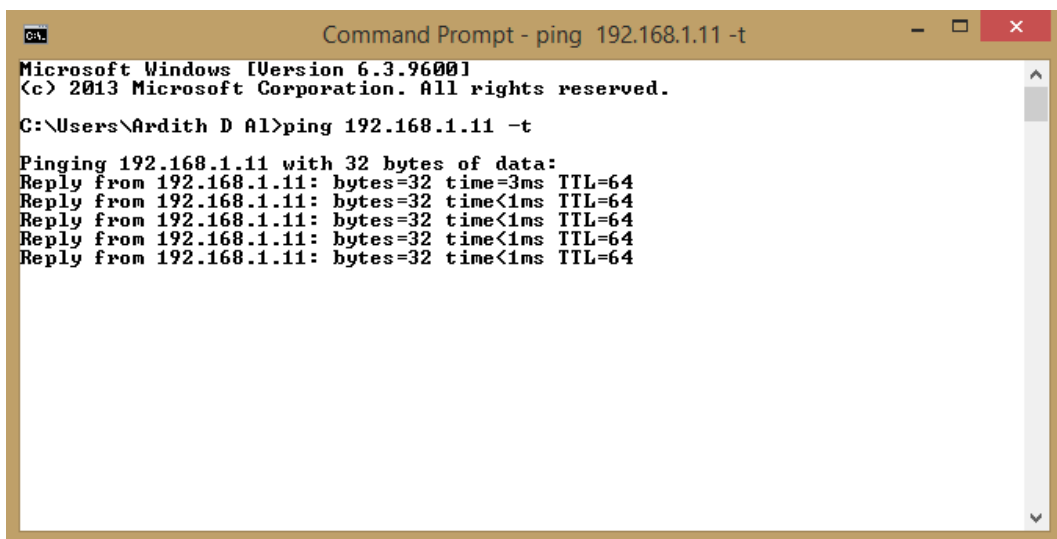
a. Tahapan Pengujian

Pada GUI *server 2* bisa dilihat *service container* Debian 7 berjalan pada *node 1* atau *server 1* seperti pada Gambar 4.



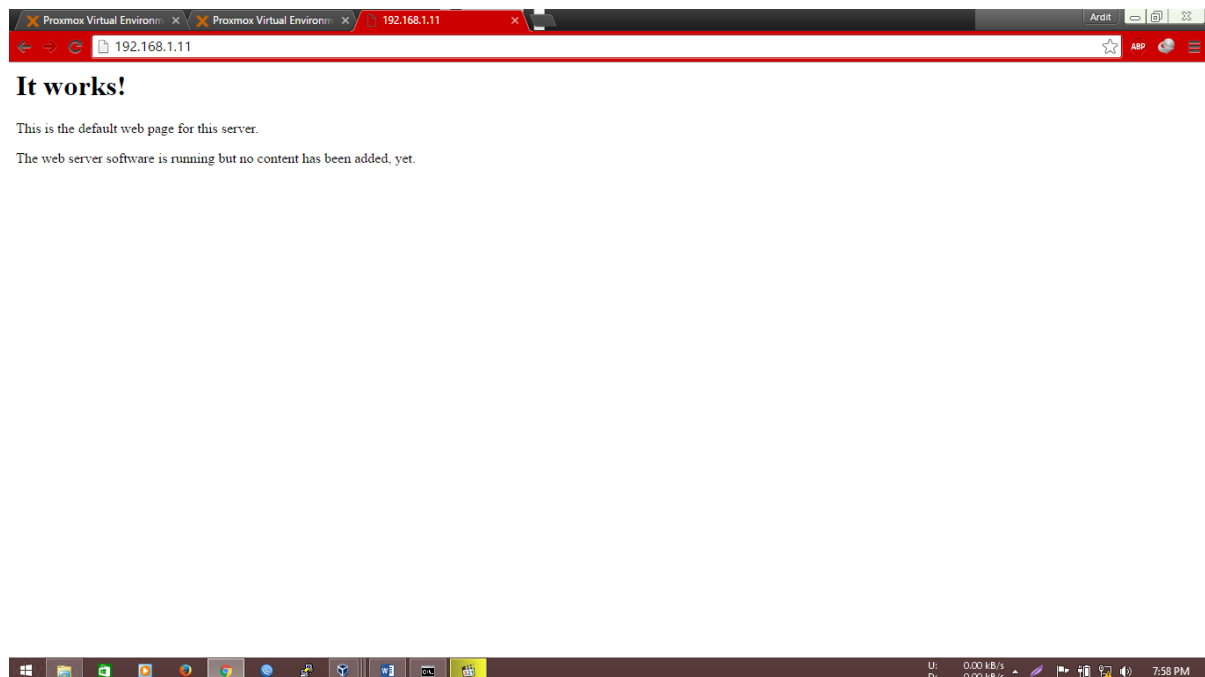
Gambar 4. GUI *server 2*

Mengirim paket atau ping ke *service* debian yang berjalan pada *server 1* untuk mengecek *service* berjalan atau belum.



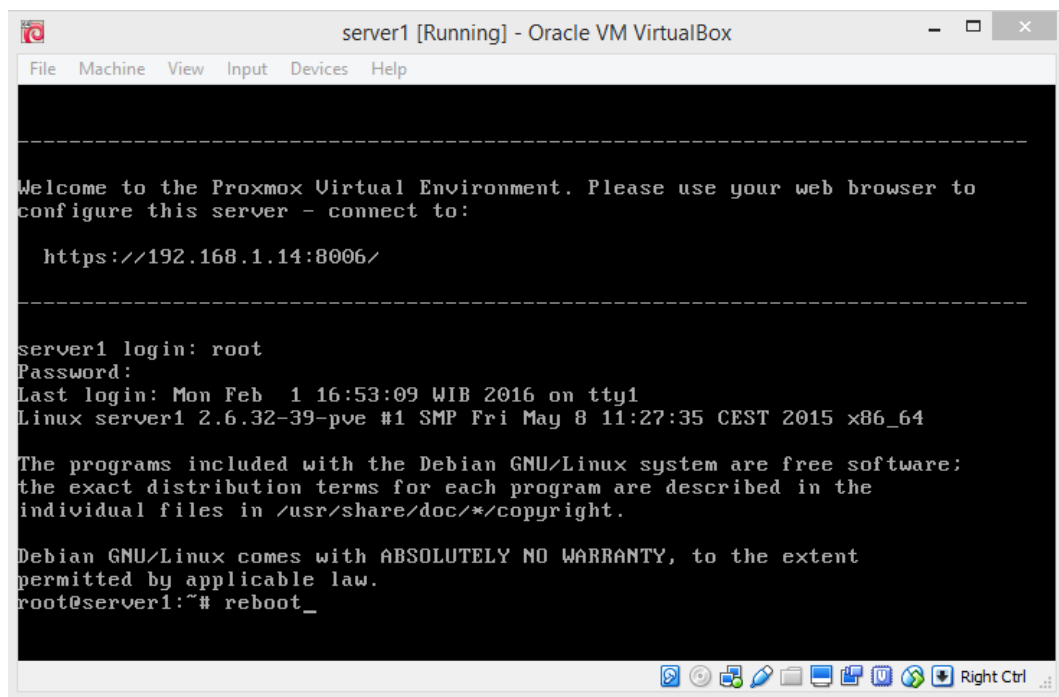
Gambar 5. Ping pada *service* yang berjalan pada *server 1*

Mengecek *web server* dalam keadaan aktif atau *down*.



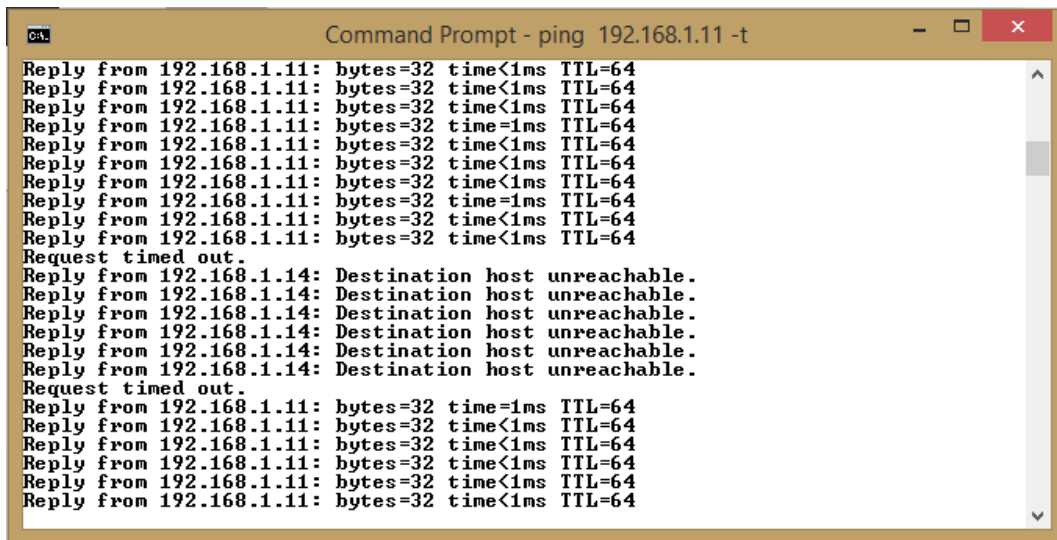
Gambar 6. *Web Server* dalam keadaan aktif

Kemudian tahap selanjutnya adalah dengan men-*shutdown server* 1 atau me-*reboot server* 1.



Gambar 7. *Reboot server* 1

Mengamati paket yang dikirim ke *service* debian mengalami *Request Time Out* dan terhubung lagi.

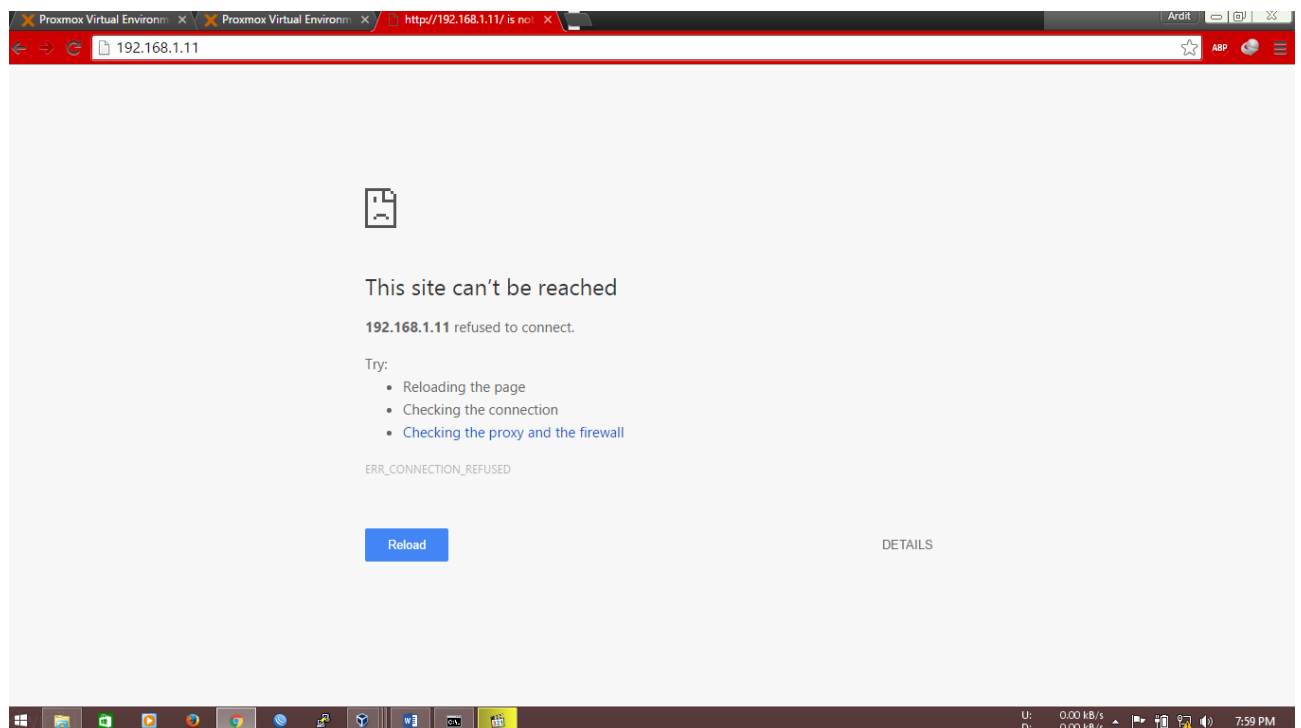


```
Command Prompt - ping 192.168.1.11 -t
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Request timed out.
Reply from 192.168.1.14: Destination host unreachable.
Reply from 192.168.1.14: Destination host unreachable.
Reply from 192.168.1.14: Destination host unreachable.
Reply from 192.168.1.14: Destination host unreachable.
Reply from 192.168.1.14: Destination host unreachable.
Request timed out.
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
Reply from 192.168.1.11: bytes=32 time<1ms TTL=64
```

Gambar 8. Ping pada *service* yang berjalan

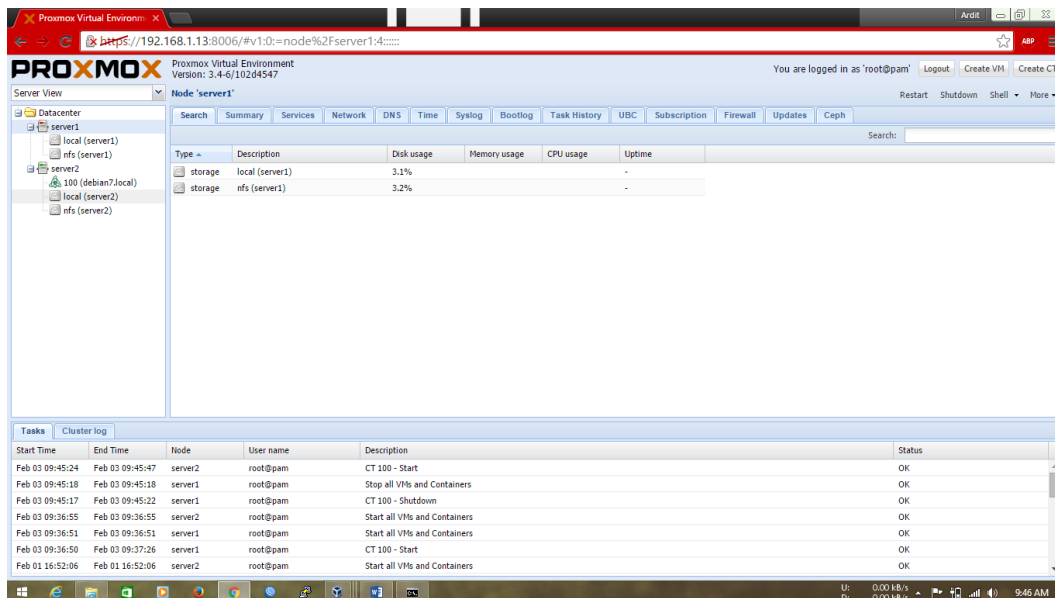
Dari Gambar 8 tersebut ping yang dilakukan mengalami RTO beberapa saat yaitu ketika *service* yang berjalan berpindah pada *node* yang aktif yang secara otomatis *service* tersebut akan melakukan *reboot*.

Mengecek kembali *web server* saat terjadi *failover*



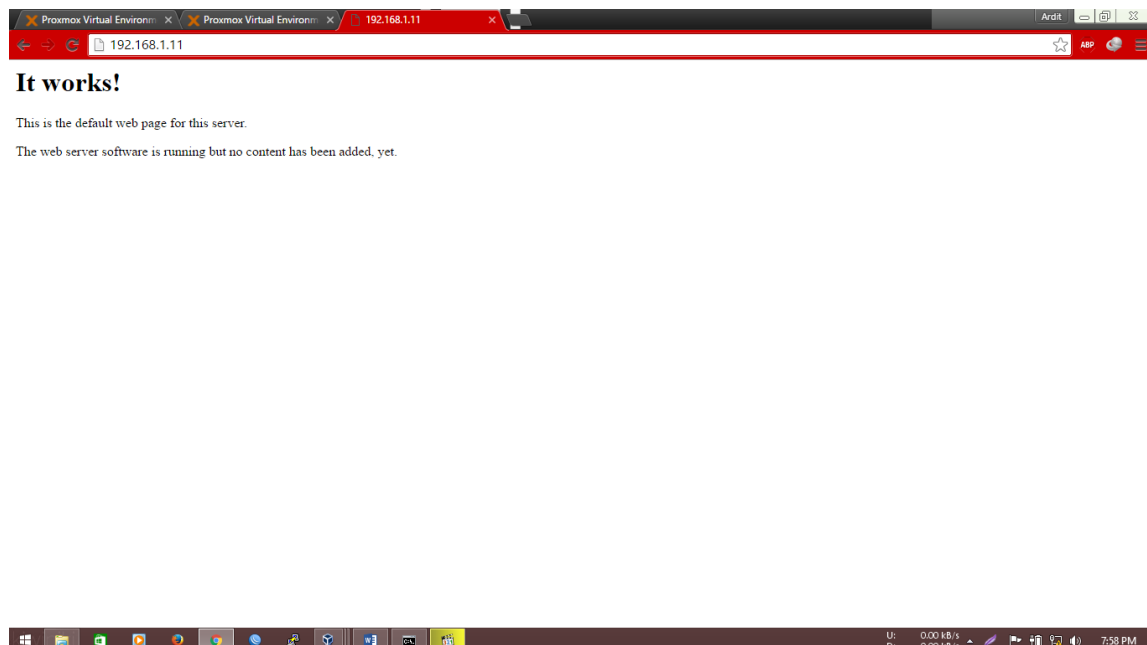
Gambar 9. Web Server down ketika terjadi *failover*

Mengamati *service* akan berpindah pada *node* yang aktif yaitu *server 2*.



Gambar 10. *Service* telah berpindah pada *node 2* atau *server 2*

Setelah *service* berpindah pada *server* yang aktif kemudian dilakukan pengecekan kembali *web server* apakah benar-benar sudah aktif kembali.



Gambar 11. *Web Server* aktif kembali

3.2 Pembahasan

- a. Availability didefinisikan sebagai probabilitas server beroperasi pada waktu yang diinginkan (Garg and Singh, 2011). Penghitungan *availability cluster* menggunakan rumus sebagai berikut :

$$Availability : \frac{MTBF}{MTBF + MTTR}$$

- MTBF (*Mean Time Before Failure*) adalah waktu rata-rata *uptime server*.
- MTTR (*Mean Time to Repair*) adalah waktu rata-rata yang diperlukan untuk mengembalikan layanan

Untuk MTBF (*Mean Time Before Failure*) didapat dengan cara menghitung waktu lama *uptime server* sampai mengalami kegagalan atau *down*. Dan untuk MTTR (*Mean Time to Repair*) didapat dengan cara menghitung waktu mengembalikan layanan dari *server 1* yang mati kemudian layanan mati dan berpindah pada *server 2* dan layanan mulai hidup kembali untuk melayani *client*. *Availability* biasanya ditentukan dengan notasi 9, dengan tingkat paling tinggi adalah 99,9% (3 sembilan).

Hasil pengujian *availability* dapat dilihat pada Tabel 3 berikut ini.

Tabel 3. Hasil Pengujian Sistem

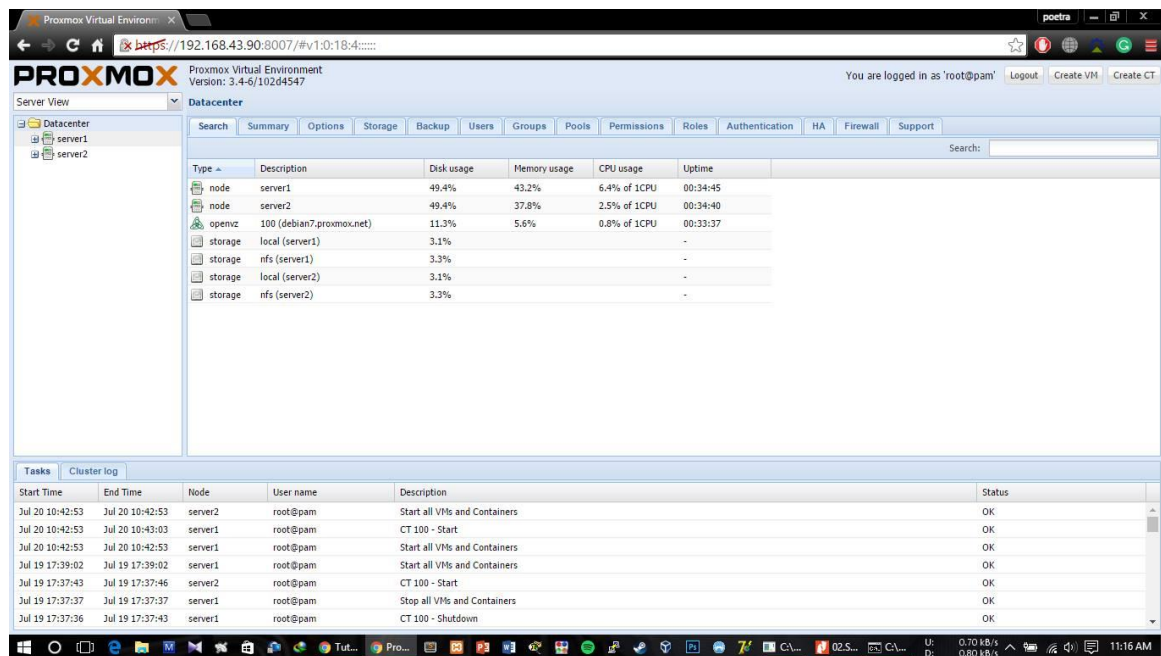
No	Tanggal	MTBF (menit)	MTTR (menit)	Availability
1	19 Juli 2016	114	0.6	99.47 %
2	19 Juli 2016	279	0.1	99.96 %
3	19 Juli 2016	99	0.1	99.89 %
4	19 Juli 2016	46.38	1.45	96.96 %
5	19 Juli 2016	25	0.28	95.12 %
6	19 Juli 2016	11.3	0.31	97.32 %
7	19 Juli 2016	16	1.3	92.48 %
8	19 Juli 2016	27	0.3	98.90 %
9	20 Juli 2016	18	0.31	98.30 %
10	20 Juli 2016	20	1.41	93.41 %
11	20 Juli 2016	52	0.31	99.40 %
12	20 Juli 2016	90	0.3	99.66 %
13	20 Juli 2016	21.19	0.31	98.55 %

14	20 Juli 2016	40	0.31	99.23 %
15	20 Juli 2016	33	0.28	99.15 %
16	20 Juli 2016	47	0.3	99.36 %
17	21 Juli 2016	77	0.61	99.21 %
18	21 Juli 2016	40	0.16	99.60 %
19	21 Juli 2016	109	0.13	99.88 %
20	21 Juli 2016	70	0.15	99.78 %
21	21 Juli 2016	92	0.18	99.80 %
22	21 Juli 2016	50	0.15	99.70 %
23	21 Juli 2016	126	0.13	99.89 %
24	21 Juli 2016	51	0.16	99.68 %
25	21 Juli 2016	37	0.13	99.64 %
Nilai Total Availability		1590.87	10.77	99.32 %
Standar Deviasi Data		56	0.43	99.22 %

Berdasarkan Tabel 3 diatas untuk analisa pengujian *availability* didapatkan hasil yang hampir mendekati *zero downtime*. Dalam pengujian yang dilakukan sebanyak 25 kali didapat rata-rata *delay* waktu 0.39 menit dengan tingkat *availability* total mencapai 99.32 %.

b. Pengujian akses *server* pada *device* lain

Dalam penelitian ini sistem berjalan pada jaringan *local*, untuk dapat diakses dari *device* lain penulis menggunakan *setting* RouterOS yang di-*install* pada VirtualBox. IP *server* yang berjalan akan di *forward* menggunakan *IP Public* agar bisa diakses dari *device* lain. Hasilnya dapat dilihat pada Gambar 12 berikut ini.



Gambar 12. Server yang berjalan diakses pada device lain

4. PENUTUP

A. Kesimpulan

Setelah melakukan serangkaian penelitian, kesimpulan yang dapat diambil berdasarkan hasil penelitian ini adalah sebagai berikut :

1. Setelah dilakukan pengujian sistem didapatkan hasil yang hampir mendekati *zero downtime*. Dalam pengujian yang dilakukan sebanyak 25 kali didapat rata-rata *delay* waktu 0.39 menit dengan tingkat *availability* total mencapai 99.32 %.
2. Untuk dapat diakses dari *device* lain penulis menggunakan *setting* RouterOS yang di-*install* pada VirtualBox. IP *server* yang berjalan akan di *forward* menggunakan *IP Public* agar bisa diakses dari *device* lain.
3. Dengan menggunakan teknologi *share storage* FreeNAS data dapat terpusat menjadi satu *server* fisik yang berbeda sehingga apabila salah satu *server* mengalami *down* maka data masih bisa diakses oleh *server* yang lain.

B. Saran

Dalam teknologi komputer khususnya *server*, tingkat *availability server* merupakan faktor yang sangat penting. Sehingga apabila *server* mempunyai tingkat *availability* yang tinggi *server* tersebut dapat terus melayani kebutuhan *client*. Admin perlu memperhatikan hal ini karena apabila *server* mengalami *down* karena suatu sebab atau *maintenance*, *client* masih dapat bisa

dilayani. Dengan menggunakan teknologi *High Availability* dapat sangat membantu dalam meningkatkan tingkat *availability server* dan meminimalisir *downtime server*.

Ketersediaan data juga mempengaruhi tingkat *availability server*. Apabila *server down* maka otomatis data yang tersimpan dalam *server* tersebut tidak dapat diakses oleh *server backup*. Dengan teknologi *share storage* memudahkan dalam penyimpanan data karena data terpusat dalam satu *server* fisik yang berbeda dan apabila salah satu *server down* maka *server* lain masih dapat mengakses data tersebut.

DAFTAR PUSTAKA

- Garg, Ritu; Singh, Awadesh Kumar. *Fault Tolerance In Grid Computing: State Of The Art And Open Issues*. International Journal Of Computer Science & Engineering Survey (IJCSSES), Vol. 2, No.1, February 2011.
- Irfani, 2015. “*Implementasi High Availability Server dengan Teknik Failover Virtual Computer Cluster*”. Skripsi. Fakultas Komunikasi dan Informatika. Universitas Muhammadiyah Surakarta.
- Thamrin, Husni. “*Perancangan Tools Berbasis Python untuk Memantau Keaktifan Server*”. Jurnal, Vol 2, No. 2, January 2011.